

TS-518E

***SAS/ACCESS® Guidelines for Connecting to
SYBASE® Databases in the UNIX® Environment***

***Technical Support Division
UNIX Systems Interface***

***SAS Institute, Inc.
SAS Campus Drive
Cary, N.C. 27513***

Abstract

Due to the increased interest of the SAS/ACCESS Interface to SYBASE product in the UNIX environment, this topic has been a popular discussion for a number of SAS users. This paper documents the necessary steps and clarifies the issues involved with connecting to SYBASE databases using SAS/ACCESS on UNIX platforms.

Introduction

The SAS/ACCESS Interface to SYBASE product provides support for the SQL Procedure Pass-Through Facility, the ACCESS Procedure and the DBLOAD Procedure.

The SAS/ACCESS product, in general, consists of three procedures: PROC ACCESS, PROC DBLOAD and PROC SQL Pass-Through. However, all three of these procedures may not be available for every relational database system (DBMS) on every UNIX platform. Users should be familiar with what procedures their SAS/ACCESS product supports for their platform. An addendum has been included for the SAS/ACCESS Interface to SYBASE support.

The ISQL utility is available with SYBASE to provide users the capability to retrieve and create data stored in the SYBASE database. Users or the Database Administrator (DBA) should be familiar with this utility because it can make the SAS/ACCESS connection process much easier.

The discussion, necessary steps, and examples that follow were run on an HP-UX system using SAS 6.11 TS020, and demonstrate the use of the SYBASE ISQL utility and successful connections to databases using SAS/ACCESS.

Using the SYBASE ISQL Utility

In order to connect to a SYBASE database locally or remotely from SAS, you must be able to connect to the same database outside of SAS. You can establish a connection outside of SAS by using the SYBASE ISQL utility.

*Once a successful database connection is made using the ISQL utility, you should be able to connect to the same database from within SAS with the **same** environment and userid. Below are some simple steps to verify whether you are able to connect to a SYBASE database outside of SAS successfully.*

- 1) *Invoke the SYBASE ISQL utility as follows, where `usr_name`, and `svr_name` are valid values for connecting to the database. If you do not know what these values are or should be or have problems connecting with the values you have specified, you will need to contact your DBA or SYBASE Technical Support for assistance. If you are not able to connect to SYBASE using the ISQL utility with the **exact** syntax demonstrated below, you will not be able to use SAS/ACCESS to connect to the database either.*

```
unix_prompt> isql -S svr_name -U usr_name
```

- 2) *You will receive the following prompt for a password and you must enter a valid password. If you have problems with your password, contact your DBA for assistance.*

```
Password: password
```

- 3) *Once you successfully connect to the database specified, you will receive the following prompt. If you are having problems connecting to the database, you will need to contact your DBA or SYBASE Technical Support for assistance.*

```
1>
```

- 4) *To verify that you have been granted the appropriate privileges, submit an SQL query from within ISQL, similar to what you plan to submit from within SAS. If you experience problems with privileges or submitting SQL from ISQL, you will need to contact your DBA or SYBASE Technical Support for assistance. Below is a simple SQL `select` statement and its results (where `tbl_name` is a valid table name in the database that you are connected):*

```
1> select * from tbl_name
2> go
```

EMPID	SALARY
-----	-----
123456	50000.00
789012	75000.00
555555	12000.00

```
(3 rows affected)
```

- 5) *You have just demonstrated that you can connect to the SYBASE database outside of SAS. Before you attempt to bring up SAS and try to connect using SAS/ACCESS, be sure that you use the **exact** `usr_name`, `password`, `svr_name` and `tbl_name` that you used with the SYBASE ISQL utility.*

Using SAS/ACCESS Interface to SYBASE

Before you attempt to connect to a SYBASE database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the SYBASE ISQL utility as demonstrated above. Once you have verified that you can connect to the database(s) using the ISQL utility, you should be able to connect to the same database(s) from within SAS, assuming the following requirements have been met.

- You are invoking SAS using the **exact** environment and userid used to invoke the ISQL utility and to connect to the database(s) successfully.
- You have successfully performed the SAS/ACCESS Interface to SYBASE Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".
- Make sure that the !SASROOT/sasexe/dbi/sassyb module, along with all other SYBASE related modules in this directory, have read and execute permissions. For dual SAS installations on SUN systems, this module can be found in !SASROOT/solaris2/sasexe/dbi for Solaris 2 platforms.
- For Solaris 2 platforms, make sure that you add the location of the SYBASE shared libraries to your systems \$LD_LIBRARY_PATH environment variable before executing the SAS/ACCESS Interface to SYBASE and SQL Server. The following syntax can be used to set the variables:

Bourne Shell:

```
LD_LIBRARY_PATH= $SYBASE/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

C Shell:

```
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$SYBASE/lib
```

Once the above requirements have been met, you should be able to successfully connect to the SYBASE database(s) using the following steps:

- 1) Invoke SAS with the **exact** environment and userid you used when you successfully connected to the database(s) using the SYBASE ISQL utility.
- 2) To verify that you are able to successfully connect to a SYBASE database, submit the following PROC SQL Pass-Through code from the SAS Program

Editor. Be sure the `usr_name`, `password`, `svr_name` and `tbl_name` are the same values used with the SYBASE ISQL utility.

```
proc sql;
connect to sybase(user="usr_name" password="password"
                  server="svr_name");
    select * from connection to sybase
        (select * from "tbl_name");
disconnect from sybase;
quit;
```

NOTE: SYBASE and the SQL Server are very case sensitive, therefore, it is highly recommended that you include the `usr_name`, `password`, `svr_name`, `db_name` and `tbl_name` in quotes (using the exact spelling, uppercase and lowercase letters as you did when using the ISQL utility). Therefore, SAS will not uppercase the values but just pass it to SYBASE.

- 3) *If, when using SAS/ACCESS, you are unable to connect to the same SYBASE database from the **exact** environment and userid that you were able to successfully connect to using the ISQL utility, then contact SAS UNIX Technical Support at (919) 677-8008 and have the following information available:*
 - *the exact operating system release level and machine type*
 - *the exact SYBASE release level*
 - *is the SYBASE database local or remote?*
 - *the environment variable settings*
 - *the results from the steps documented above for both "Using the SYBASE ISQL Utility" and "Using SAS/ACCESS."*
- 4) *Once you have connected successfully and want more details on using the ACCESS Procedure, DBLOAD Procedure or the SQL Procedure Pass-Through Facility, reference the SAS manuals listed in the CONCLUSION of this paper.*

Conclusion

This paper discussed the various aspects of connecting to SYBASE databases with both the SYBASE ISQL utility outside of SAS and with SAS using the SAS/ACCESS Interface to SYBASE product on UNIX platforms. This discussion includes details and examples of using the ISQL utility and the SAS SQL Procedure Pass-Through Facility.

For detailed information on the SYBASE SQL commands used in this report, please refer to your SYBASE manuals or ask your Database Administrator for assistance.

For detailed information on the SAS procedures PROC ACCESS, PROC DBLOAD, PROC SQL PASS-THROUGH, reference the following manuals

SAS/ACCESS Software for Relational Databases Reference, Version 6 (# 55144)

SAS/ACCESS Interface to SYBASE and SQL Server Usage and Reference, Version 6 (# 56081)

SAS/ACCESS Software Changes and Enhancements, SQL Procedure Pass-Through Facility, Version 6 (# 55237)

SAS Software Changes and Enhancements, Release 6.11 (# 55300)

SAS Technical Report, Append Capability for PROC DBLOAD (TS-347)

Getting Started with SAS/ACCESS Software, Version 6 (# 55103)

SAS/ACCESS INTERFACE TO SYBASE SUPPORT **ADDENDUM**

6.07 PLATFORMS

<u>Platform</u>	<u>Sybase</u>
-----------------	---------------

1. SUN4	TS203
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

2. HPU8	TS210
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

3. AIXR	TS105
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

6.09 PLATFORMS

<u>Platform</u>	<u>Sybase</u>
-----------------	---------------

1. SUN4	TS027
---------	-------

2. SOL2	TS037
---------	-------

3. HPU8	TS027
---------	-------

4. AIXR	TS027
---------	-------

6.10 PLATFORMS

<u>Platform</u>	<u>Sybase</u>
-----------------	---------------

1. Alpha/OSF	N/A
--------------	-----

2. MABI	
---------	--

- SGI	TS018
-------	-------

- Pyramid	TS018
-----------	-------

** SQL passthrough only; shipped experimental and went production in 6.11 Wave 2.*

3. IABI	
---------	--

- NCR	TS018
-------	-------

- Sequent	TS018
-----------	-------

** SQL passthrough only; shipped experimental only.*

6.11 PLATFORMS

<u>Platform</u>	<u>Sybase</u>
1. SUN4 & SOL2	TS020
2. HPU8	TS020
3. AIXR	TS020
4. Digital UNIX	N/A
5. MABI	
- SGI	N/A
- Pyramid	N/A
6. IABI	
- NCR	N/A
- Sequent	N/A

6.12 BETA PLATFORMS

<u>Platform</u>	<u>Sybase</u>
1. SUN4 & SOL2	TS005
2. HPU8	TS005
3. AIXR	TS005